

Programação Visual

Trabalho de Laboratório nº 0

Objetivo	Familiarização com o ambiente de desenvolvimento Visual Studio. Criação de aplicações de consola simples. Introdução/Conversão à linguagem C# com a criação/utilização de classes, herança, polimorfismo e interfaces.
Programa	Protótipo do Jogo de Xadrez.
Regras	Criar uma aplicação de consola (.Net Core). Use as convenções de codificação adotadas para a linguagem C#. Na classe do programa não coloque atributos nem crie nenhum método para além do Main . Não é necessário obter dados do utilizador. Forneça os dados ao nível do código.

Descrição



Nível 1

- Crie a classe **Posicao** que representa uma posição do tabuleiro de Xadrez. Acrescente a esta classe duas propriedades implícitas, **X** e **Y**, para guardar, respetivamente, o valor da coordenada **X**, um caracter entre 'a' e 'h', e o valor da coordenada **Y**, um valor inteiro entre 1 e 8. Complete a classe com dois construtores, um sem parâmetros que cria um objeto nas coordenadas **a** e **1** (use constantes de classe para guardar estes valores) e o outro com dois parâmetros que recebem o valor das propriedades. Inclua o método **ToString()** que retorna um texto com os valores de **X** e **Y** concatenados (ex: "a2").
- Teste a classe criada criando dois objetos usando construtores diferentes. Escreva no ecrã a informação dos objetos criados.

Nível 2

- Construa uma hierarquia de classes com base na classe *abstrata* **Peca** que inclui as classes derivadas **Peao** e **Torre**. A classe **Peca** deve guardar a informação da sua posição e da sua cor. Para a cor use um tipo enumerado **Cor** com os valores **Branco** e **Preto**. Inclua os construtores, um sem parâmetros e o outro com dois parâmetros que recebem o valor dos atributos. Inclua ainda propriedades explícitas para os atributos definidos e o método **ToString()** que retorna a posição da peça reutilizando o método equivalente que existe na classe **Posicao**.
- As classes derivadas **Peao** e **Torre** não acrescentam atributos e ambas têm apenas um construtor que recebe a cor e a posição da peça. O **ToString()** da classe **Peao** devolve o mesmo que o **ToString()** da classe **Peca**. O **ToString()** da classe **Torre** acrescenta antes da posição a letra T maiúscula que representa a torre (ex: Ta8).
- Teste as classes criadas.

Programação Visual

Trabalho de Laboratório nº 0

Nível 3

- Inclua agora a propriedade **Nome** na classe **Peca** que retorna o texto “desconhecida”. Redefina esta propriedade nas classe **Peao** e **Torre** de forma a retornar o texto “Peão” e “Torre” respetivamente.
- Para testar o polimorfismo crie um *array* de objetos da classe **Peca** e coloque no mesmo pelo menos um objeto da classe **Peao** e outro da classe **Peca**. Pode usar os objetos que criou antes. De seguida, usando um ciclo **foreach**, percorra o *array* escrevendo no ecrã o nome de todas as peças que contem.

Nível 4

- Acrescente uma interface **IMover** para reunir as funcionalidades de movimento das pecas. O único método a definir - **void Deslocar(int dx, int dy)** - deverá mover uma peça um determinado valor no eixo do x (**dx**) e outro no eixo do y (**dy**).
- Implemente a interface **IMover** na classe **Peca**. Nesta classe o método **Deslocar** deve ser abstrato.
Nota: Por agora não precisa ter em conta valores inválidos para o movimento, assuma apenas que o peão não se move na horizontal e a torre só aceita um dos movimentos (**dx** ou **dy**). Se forem ambos diferentes de zero não se move.
- Crie as classes que faltam para as peças de xadrez: **Cavalo**, **Bispo**, **Rei** e **Rainha**. Use como símbolos das peças nestas classes as letras **C**, **B**, **R** e **D** respetivamente. Não coloque código nos métodos **Deslocar** destas classes.
- Teste.

Nível 5

- Crie a classe **Tabuleiro** que usa um *array* bidimensional para guardar as peças. Implemente nesta classe o construtor sem parâmetros que inicializa o tabuleiro com as peças normais do xadrez nas suas posições iniciais.
- Crie um método **Mostrar()** na classe **Tabuleiro** que escreve para o ecrã o tabuleiro com as suas peças. Use os símbolos das peças para as representar e inclua as coordenadas.
- Teste.

Desafio

- Crie **Indexers** para o tabuleiro de forma a que possa aceder diretamente às posições do tabuleiro a partir do método **Main**.
- Teste.

Notas

- Para os identificadores siga as convenções adotadas pelo C#, nomeadamente:
- A notação camelCase para o nome das variáveis locais e identificadores privados.
 - A notação PascalCase para os nomes públicos dos métodos, classes e interfaces.
 - Não utilize o símbolo ‘_’ nos identificadores nem abreviaturas